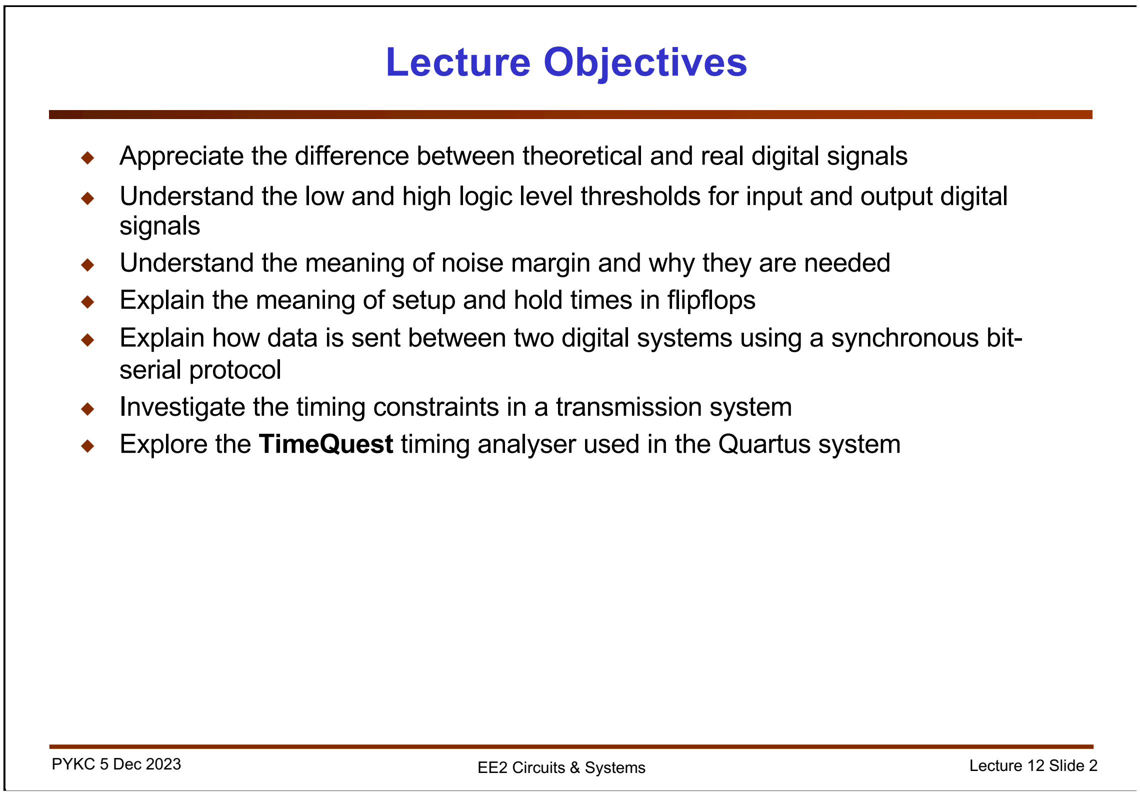
Imperial College
London

Lecture 12

Timing Constraints & Timing Analysis

Peter Cheung
Imperial College London

URL: www.ee.imperial.ac.uk/pcheung/teaching/E2_CAS/
E-mail: p.cheung@imperial.ac.uk



Lecture Objectives

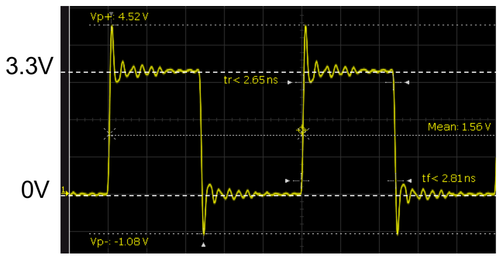
- Appreciate the difference between theoretical and real digital signals
- Understand the low and high logic level thresholds for input and output digital signals
- Understand the meaning of noise margin and why they are needed
- Explain the meaning of setup and hold times in flipflops
- Explain how data is sent between two digital systems using a synchronous bit-serial protocol
- Investigate the timing constraints in a transmission system
- Explore the **TimeQuest** timing analyser used in the Quartus system

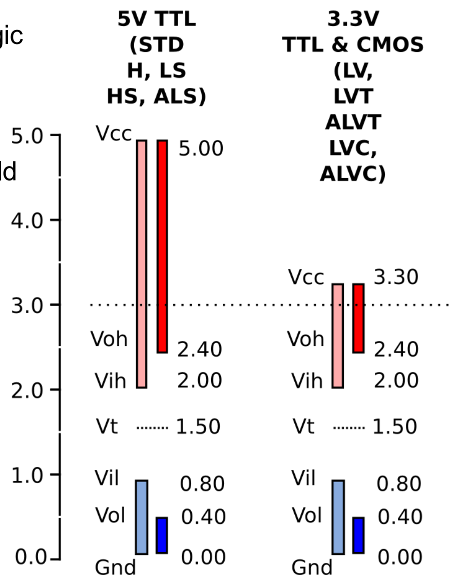
PYKC 5 Dec 2023 EE2 Circuits & Systems Lecture 12 Slide 2

In this lecture, we will first examine practical digital signals. Then we will discuss the timing constraints in digital systems. The important concepts are related to **setup** and **hold times** of registers and how these, together with delay time of combinational circuit, determine how fast a digital system could run at.

Typical digital signal

- Real digital signals are generally far from ideal.
- Shown here is a 4MHz digital signal using 3.3V logic as measured on a digital oscilloscope.
- There are overshoots and undershoots in voltage levels and finite rise and fall times.
- That's why logic circuits have well-defined threshold voltages for high and low levels as shown on the right.
- For 3.3V logic, Voh ≥ 2.4V and Vih ≥ 2V, therefore the high level margin (noise margin) is 0.4V





Shown here is a digital signal produced by an ARM microcontroller as measured with a digital oscilloscope. This ARM microcontroller uses the 3.3V logic standard, the same as we use with the DE10-Lite board in Lab-in-a-Box. The waveform has both overshoots and undershoots immediately after the rising and falling transitions. Part of the overshoots are due to the scope probe (and the inductance in the ground lead). However, even on-chip digital signals have some degree of overshoots. Furthermore, there could also be spurious signals (i.e. noise) coupled onto any digital signals.

Fortunately, digital signals are characterised as low ('0') or high ('1') by threshold voltages. Shown on the right are the digital thresholds defined for 5V TTL logic and 3.3V logic.

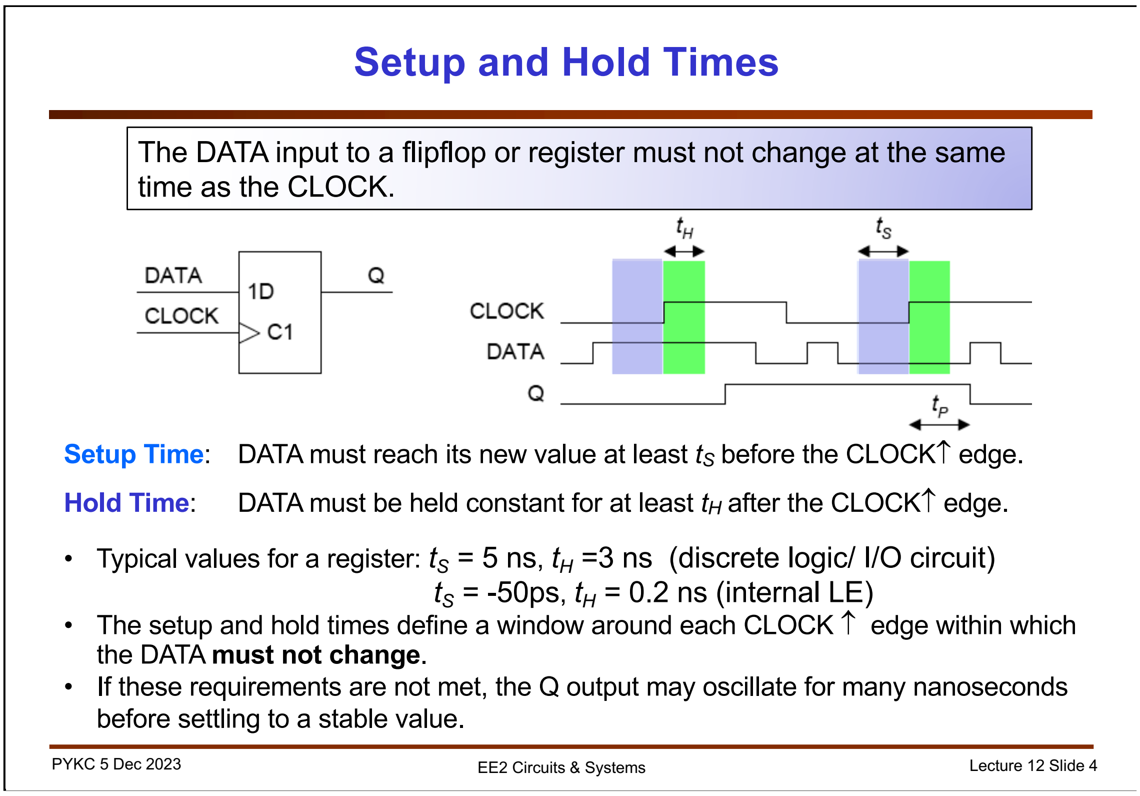
Let us consider the high logic level for 3.3V logic. Two threshold voltages are defined:

Voh = output high threshold voltage – all logic circuits with a high output will drive a circuit node at 2.4V or higher.

Vih = input high threshold voltage – all logic circuits will regard an input voltage as high ('1') if it is 2V or higher.

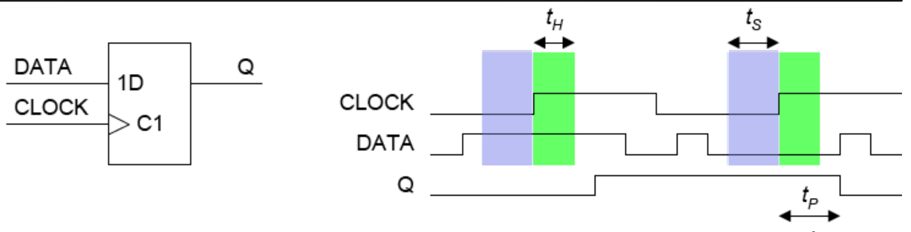
The difference Voh – Vih = 0.4V is the margin of error between the driving circuit and the input circuit. It is called the **noise margin**. It is the amount of overshoot, undershoot or noise that could be tolerated on a digital signal wire without it being interpreted wrongly by the circuit.

Note that 3.3V logic is actually compatible with 5V TTL logic (i.e. they have the same threshold voltages). Most 3.3V input pins are 5V tolerant, meaning that it can withstand a signal up to 5V without damaging the internal circuit.



Setup and Hold Times

The DATA input to a flipflop or register must not change at the same time as the CLOCK.



Setup Time: DATA must reach its new value at least t_S before the CLOCK↑ edge.

Hold Time: DATA must be held constant for at least t_H after the CLOCK↑ edge.

- Typical values for a register: t_S = 5 ns, t_H =3 ns (discrete logic/ I/O circuit)
  t_S = -50ps, t_H = 0.2 ns (internal LE)
- The setup and hold times define a window around each CLOCK ↑ edge within which the DATA **must not change**.
- If these requirements are not met, the Q output may oscillate for many nanoseconds before settling to a stable value.

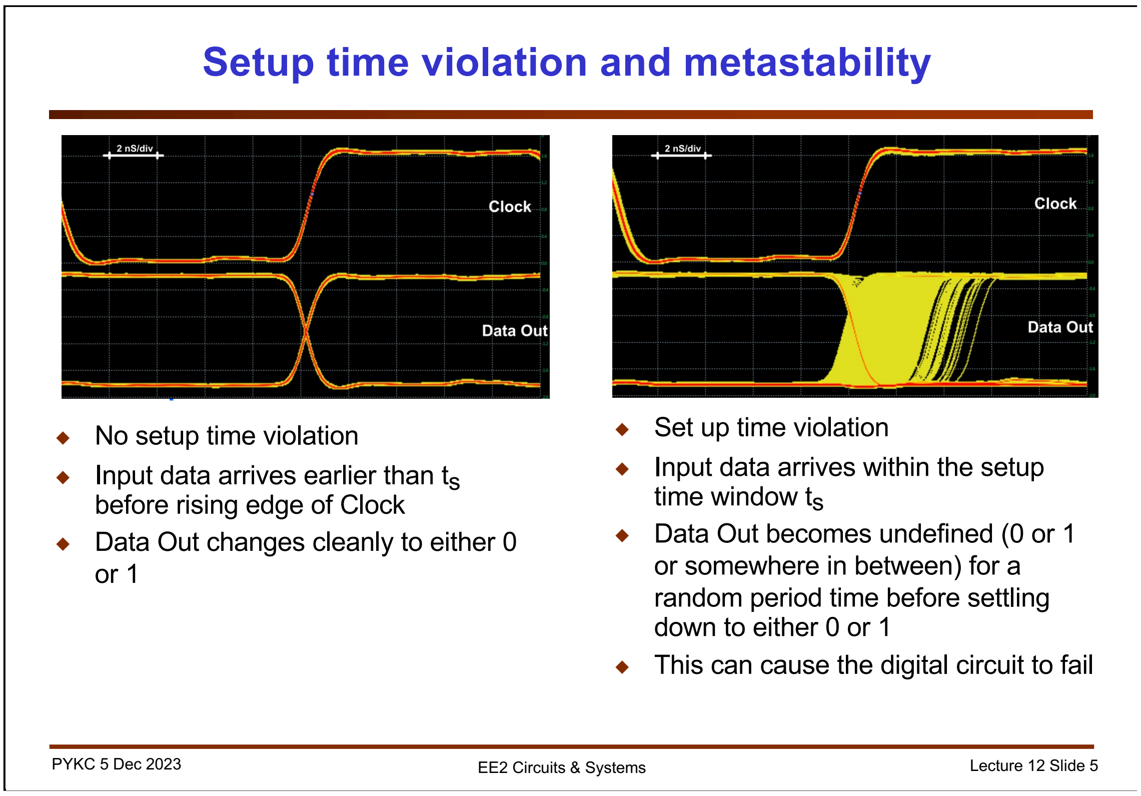
PYKC 5 Dec 2023 EE2 Circuits & Systems Lecture 12 Slide 4

Registers (D-FFs) are used everywhere in digital circuits. Using registers has the advantage of: 1) **synchronising** all activities to a clock signal; 2) **isolate** different part of the digital systems between registers (because the registers block the signal until the next active edge of the clock; 3) makes timing consideration much easier to handle.

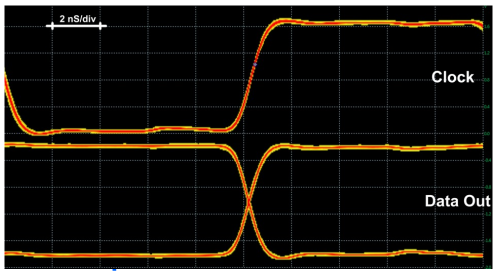
In the circuit shown here, the D-flipflop is triggered on the rising edge of the clock. The value in DATA is sampled and stored, and keep as output Q. However, for reliable operations, DATA MUST BE STABLE some time before the rising edge of CLOCK. This time is known as **setup time** t_S. This time is needed because there is internal propagation of the DATA signal which must be taken into account. As a result, for the D-flipflop to work, such internal delay is specified as the flipflop setup time requirement.

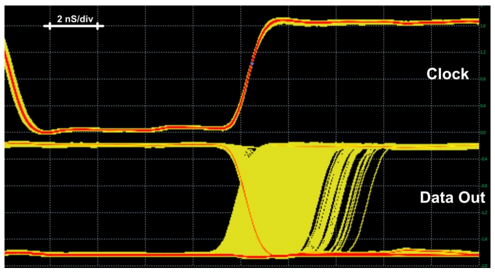
Similarly, DATA MUST BE STABLE and holds its value some time after the rising edge of CLOCK. This time is known as hold time t_H.

What happens if data changes within the setup/hold time window? The Q output becomes unknown (could be ‘1’ or ‘0’, or at a voltage level that is between the two). Eventually Q will go to ‘0’ or ‘1’, but the time it takes to reach the stable Q value is random! Such a state of the flipflop is known as a “**metastable**” state.



Setup time violation and metastability





- No setup time violation
- Input data arrives earlier than t_S before rising edge of Clock
- Data Out changes cleanly to either 0 or 1

- Set up time violation
- Input data arrives within the setup time window t_S
- Data Out becomes undefined (0 or 1 or somewhere in between) for a random period time before settling down to either 0 or 1
- This can cause the digital circuit to fail

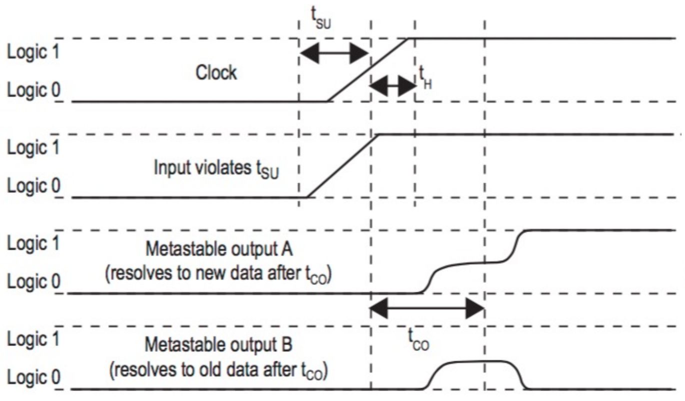
PYKC 5 Dec 2023 EE2 Circuits & Systems Lecture 12 Slide 5

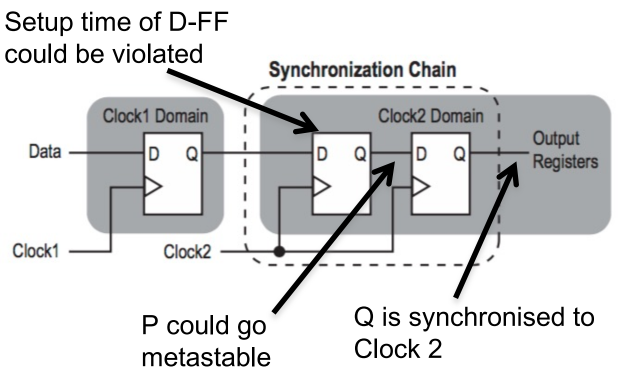
The waveforms shown here illustrates what happens when setup time violation occurs. The Data Out signal becomes indeterminate for a period of time before settling down to either 0 or 1.

Why would this cause circuit to fail. This metastable logic signal could be captured by two different D-FFs, one could resolve its output A to '1', and another could resolve its output B to '0'. Therefore the same logic signal could be interpreted by the circuit as two different logic values.

Metastability is a problem that arises when an external input NOT synchronised to the system clock is fed into our synchronous circuit. Since the input signal could change anytime relative the the clock edge, metastability will occur. It could also happens when a signal crosses from one clock domain (Clock1) to another clock domain (Clock2).

To avoid the metastable signal causing error in the digital system, one could use a synchronization chain as shown below.

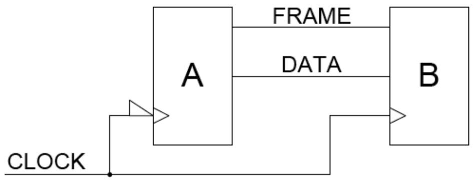


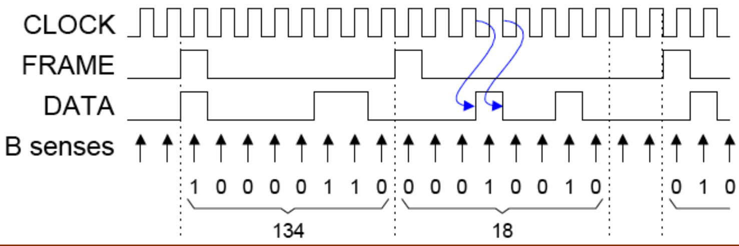


Synchronous Bit-Serial Transmission

Transmitting 8 bit values from A to B:

- FRAME indicates the first bit of each value; the other 7 bits follow on consecutive clock cycles. The FRAME signal is often called **a frame sync pulse**.
- DATA changes on the *falling* CLOCK edge
- Propagation delays are often omitted from diagram.
- DATA is sensed by system B on the *rising* CLOCK edge to maximise tolerance to timing errors. We must always clock a flipflop at a time when its DATA input is not changing.



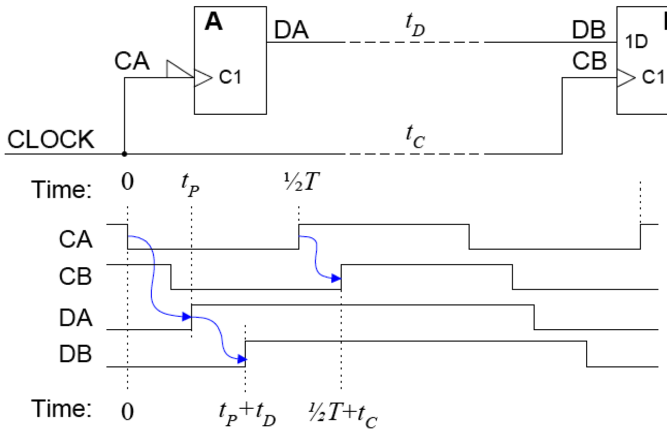


Let us consider two systems A and B, and we want to send digital data between them. The obvious method is to send the digital data one bit at a time. Such serial communication method has many advantages: 1) It is very simple to do; 2) it only needs very few wires linking between the two systems.

If the communication is governed by a clock signal, it is a synchronous bit-serial transmission system. Here we need a clock signal and a data signal. Since in most cases, we are interested in data that are more than one bit (for example, you may be interested in a block of data occupying, say, 134 bits). This block of data is known as a "**frame**". To identify when a frame of data starts, we may need another signal FRAME to indicate where the first bit starts.

In this example, the sender is triggered on the falling edge of the clock, and the receiver (at B) is triggered on the rising edge of the clock.

Timing Specifications



For Device B:

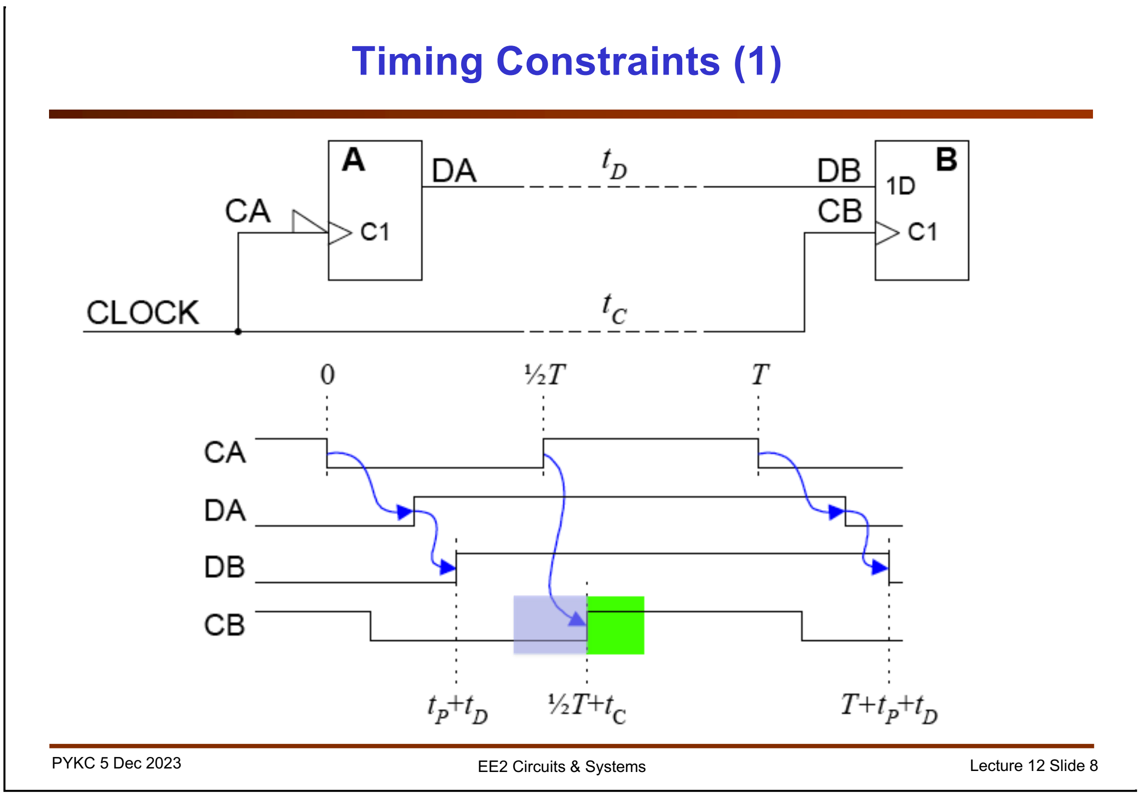
◆Data input changes at time t_P+t_D

◆Clock input changes ↑ at time $\frac{1}{2}T+t_C$

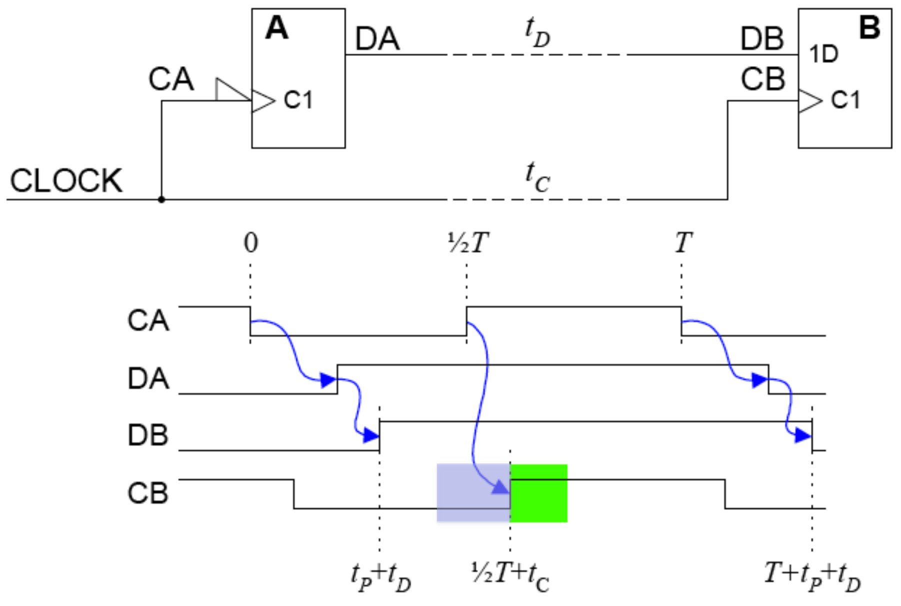
t_P	*Propagation* delay for device A.
T	Clock Period.
t_C, t_D	Transmission line delays for CLOCK and DATA

Here is the timing diagram for the data travelling from A to B via a synchronous serial link. CA is the clock signal to module A. It also supply CB, but CB is delayed by t_C due to the propagation from A to B.

DA changes t_P after the falling edge of the clock. The propagation delay of the data signal is t_D.

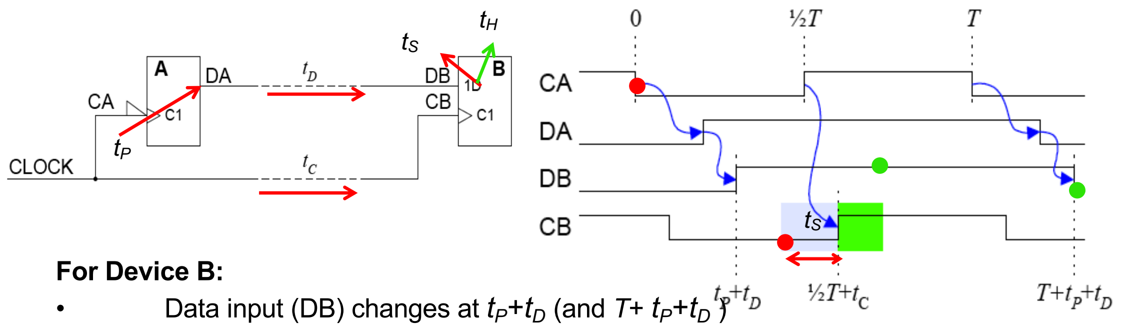


Timing Constraints (1)



In order to guarantee reliable working of the serial interface circuit, the rising edge of CB must become stable outside the setup time window (shown in light blue).

Timing Constraints (2)



For Device B:

- Data input (DB) changes at t_P+t_D (and T+ t_P+t_D)
- Clock↑ (CB) at time ½T+t_C

For reliable operation:

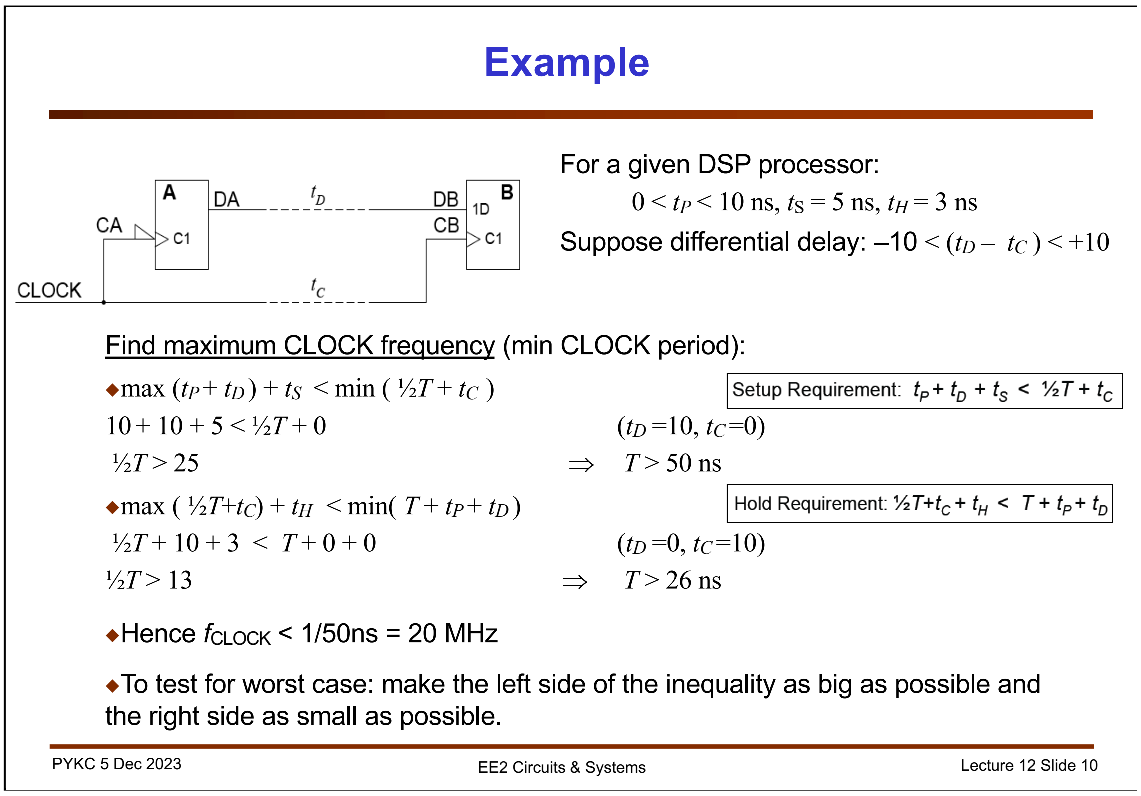
- Setup Requirement: $t_P + t_D + t_S < ½T + t_C$
- Hold Requirement: $½T+t_C + t_H < T + t_P + t_D$

Get a pair of inequalities for each flipflop/register in a circuit.
You never get both t_S and t_H in the same inequality.

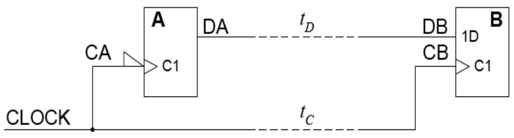
In order to consider the timing constraints for this circuit, we only need to focus on the receiving FF B. We ask the questions:

1.When DB is sampled on the rising edge of CB, is DB stable or not? The answer to this question produces the setup time requirement constraint. Here we consider what causes DB to change (the falling edge of CA), and how long it takes for this change to propagate to DB (t_P + t_D). Then we add the setup time to this (because CB MUST BE STABLE t_S before the clock edge). This must then be shorter than the time a which DB is sampled by CB. That is, this must occur on the rising edge of CB (which is ½T + t_C).

2.After DB is captured by the FF, will DB holds its value long enough? We now examine after sampling, when will DB change next. This occurs at T + (t_P + t_D), and produces the hold time constraint.



Example



For a given DSP processor:

$0 < t_P < 10$ ns, $t_S = 5$ ns, $t_H = 3$ ns

Suppose differential delay: $-10 < (t_D - t_C) < +10$

Find maximum CLOCK frequency (min CLOCK period):

$\boxed{\text{Setup Requirement: } t_P + t_D + t_S < \tfrac{1}{2}T + t_C}$

◆$\max(t_P + t_D) + t_S < \min(\tfrac{1}{2}T + t_C)$

$10 + 10 + 5 < \tfrac{1}{2}T + 0$ $\quad (t_D = 10,\ t_C = 0)$

$\tfrac{1}{2}T > 25 \quad \Rightarrow \quad T > 50$ ns

$\boxed{\text{Hold Requirement: } \tfrac{1}{2}T + t_C + t_H < T + t_P + t_D}$

◆$\max(\tfrac{1}{2}T + t_C) + t_H < \min(T + t_P + t_D)$

$\tfrac{1}{2}T + 10 + 3 < T + 0 + 0$ $\quad (t_D = 0,\ t_C = 10)$

$\tfrac{1}{2}T > 13 \quad \Rightarrow \quad T > 26$ ns

◆Hence $f_{CLOCK} < 1/50\text{ns} = 20$ MHz

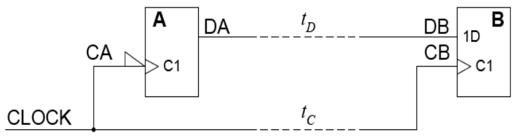
◆To test for worst case: make the left side of the inequality as big as possible and the right side as small as possible.

PYKC 5 Dec 2023 EE2 Circuits & Systems Lecture 12 Slide 10

Let us plug some numbers into the system here. Note that timing constraints such as t_P may be specified as a range of values. In this case $0 < t_P < 10$ns. You must choose the maximum value (worst case) for parameters on left side of <, and minimum value on the right side of <.

Here we can calculate the minimum period (and hence the maximum frequency) that the circuit can operate reliability without violating either the setup time or the hold time constraints.

Propagation Delay Constraint Inequalities



When do they arise?

Whenever a flipflop's clock and data input signals originate from the same ultimate source. Here CB and DB both originate from CLOCK. You normally get two inequalities for each flipflop in a circuit.

Relationship between setup and hold inequalities:

- Setup Requirement: $t_P + t_D + t_S < ½T + t_C$
- Hold Requirement: $½T + t_C + t_H < t_P + t_D + T$

IMPORTANT:
These inequalities applies ONLY to this circuit.
IT IS NOT UNIVERSAL!

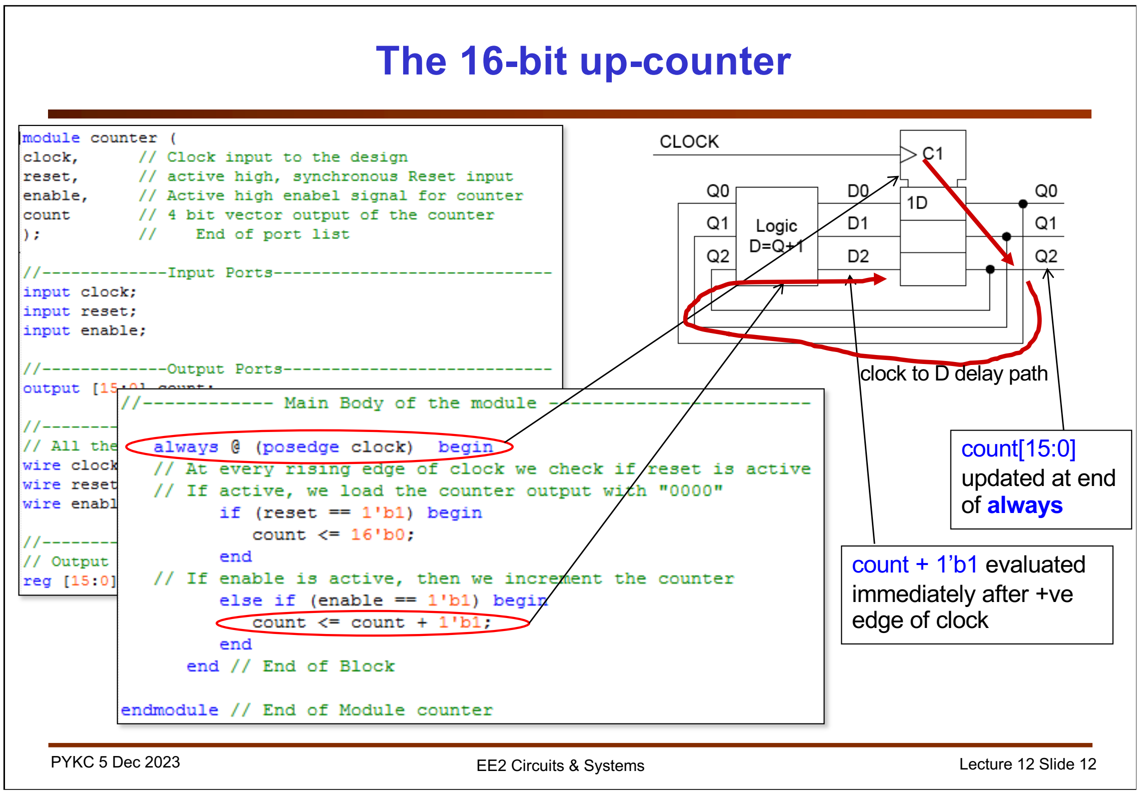
Are both t_S and t_H ever in the same inequality?

- No.

How do you decide to take the max or the min?

- For a <, take max of everything on the left and min of everything on the right.
- max = most positive: for example, max(–7,–2) = –2

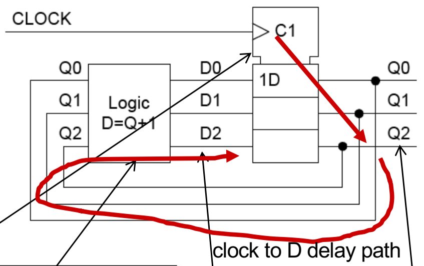
When do you need to consider these inequalities? Whenever you consider sequential circuits where the data and/or clock signals are derived from the same source.



The 16-bit up-counter

```
module counter (
clock,       // Clock input to the design
reset,       // active high, synchronous Reset input
enable,      // Active high enabel signal for counter
count        // 4 bit vector output of the counter
);           //    End of port list

//-------------Input Ports-----------------------------
input clock;
input reset;
input enable;

//-------------Output Ports----------------------------
output [15

//--------
// All the
wire clock
wire reset
wire enabl

//--------
// Output
reg [15:0]
```

```
//------------ Main Body of the module ------------------------

  always @ (posedge clock)  begin
  // At every rising edge of clock we check if reset is active
  // If active, we load the counter output with "0000"
      if (reset == 1'b1) begin
         count <= 16'b0;
      end
  // If enable is active, then we increment the counter
      else if (enable == 1'b1) begin
         count <= count + 1'b1;
      end
    end // End of Block

endmodule // End of Module counter
```

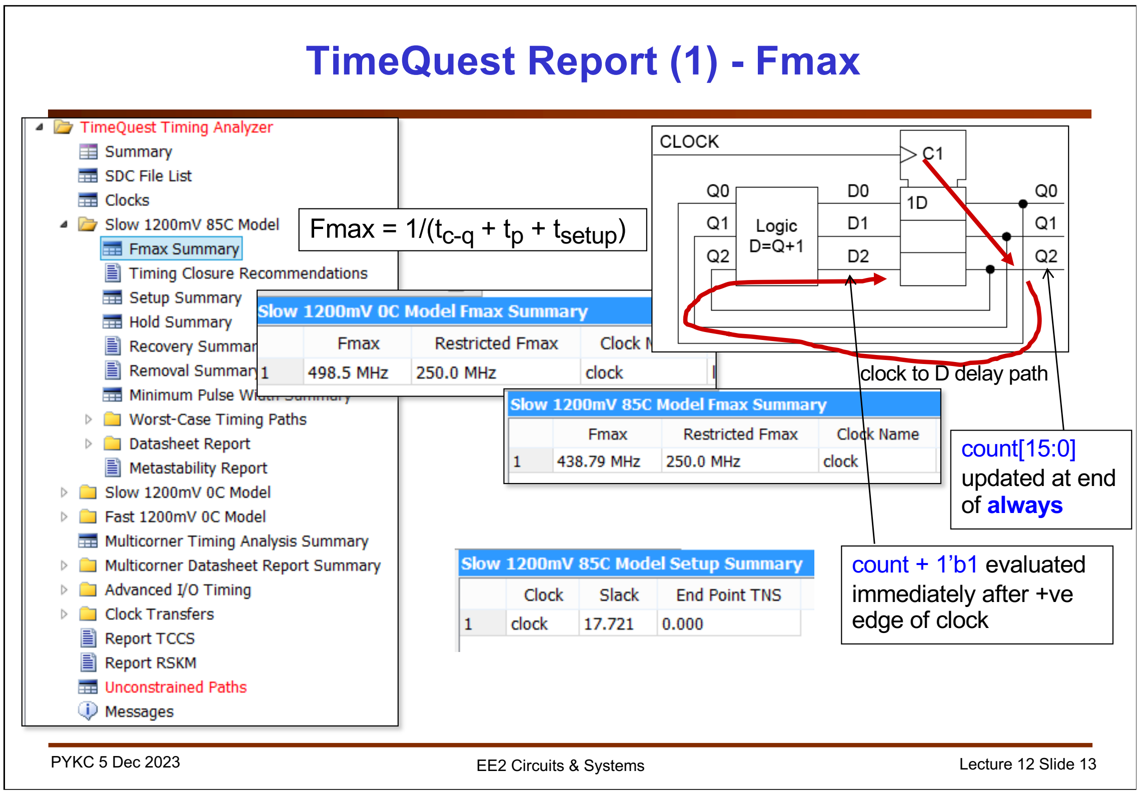


count[15:0] updated at end of **always**

count + 1'b1 evaluated immediately after +ve edge of clock

Now let us consider the Verilog specification for a 16-bit up-counter. The red arrows here indicates the delay paths from the rising edge of the clock to the Q output, then through the logic block D=Q+1 and you must also add in the setup time of the flipflop.

Note how this counter is specified in Verilog.



TimeQuest Report (1) - Fmax

- TimeQuest Timing Analyzer
  - Summary
  - SDC File List
  - Clocks
  - Slow 1200mV 85C Model
    - Fmax Summary
    - Timing Closure Recommendations
    - Setup Summary
    - Hold Summary
    - Recovery Summar
    - Removal Summar
    - Minimum Pulse Wi
    - Worst-Case Timing Paths
    - Datasheet Report
    - Metastability Report
  - Slow 1200mV 0C Model
  - Fast 1200mV 0C Model
  - Multicorner Timing Analysis Summary
  - Multicorner Datasheet Report Summary
  - Advanced I/O Timing
  - Clock Transfers
  - Report TCCS
  - Report RSKM
  - Unconstrained Paths
  - Messages

$$F_{max} = 1/(t_{c\text{-}q} + t_p + t_{setup})$$

Slow 1200mV 0C Model Fmax Summary

	Fmax	Restricted Fmax	Clock N
1	498.5 MHz	250.0 MHz	clock

Slow 1200mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name
1	438.79 MHz	250.0 MHz	clock

Slow 1200mV 85C Model Setup Summary

	Clock	Slack	End Point TNS
1	clock	17.721	0.000

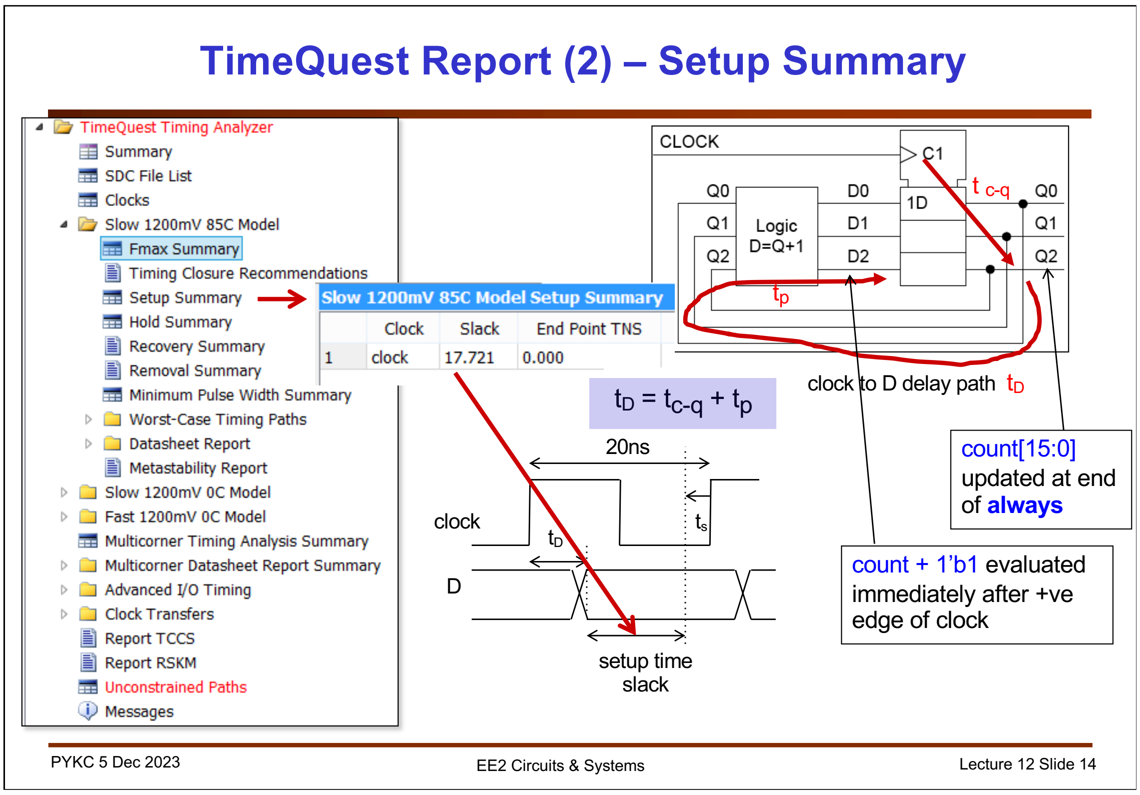
count[15:0] updated at end of **always**

count + 1'b1 evaluated immediately after +ve edge of clock

PYKC 5 Dec 2023 EE2 Circuits & Systems Lecture 12 Slide 13

If you implement this on the Cyclone III FPGA, you can use the **timing analyser**, known as **TimeQuest**, in Quartus to work out the timing constraints for you. This reports that the maximum operating frequency of the counter is 498.5MHz. However, due to the limitations of the pins, the maximum observable frequency is 250MHz. This is because the pins and pads of the chip is rather slower than the internal logic.

For this year, we have upgraded to a MAX10 FPGA, and the actually frequency of the counter will be different than that shown here. However, the underpinning principle is the same.



TimeQuest Report (2) – Setup Summary

- TimeQuest Timing Analyzer
  - Summary
  - SDC File List
  - Clocks
  - Slow 1200mV 85C Model
    - Fmax Summary
    - Timing Closure Recommendations
    - Setup Summary
    - Hold Summary
    - Recovery Summary
    - Removal Summary
    - Minimum Pulse Width Summary
    - Worst-Case Timing Paths
    - Datasheet Report
    - Metastability Report
  - Slow 1200mV 0C Model
  - Fast 1200mV 0C Model
  - Multicorner Timing Analysis Summary
  - Multicorner Datasheet Report Summary
  - Advanced I/O Timing
  - Clock Transfers
  - Report TCCS
  - Report RSKM
  - Unconstrained Paths
  - Messages

Slow 1200mV 85C Model Setup Summary

	Clock	Slack	End Point TNS
1	clock	17.721	0.000

$t_D = t_{c\text{-}q} + t_p$

count[15:0] updated at end of **always**

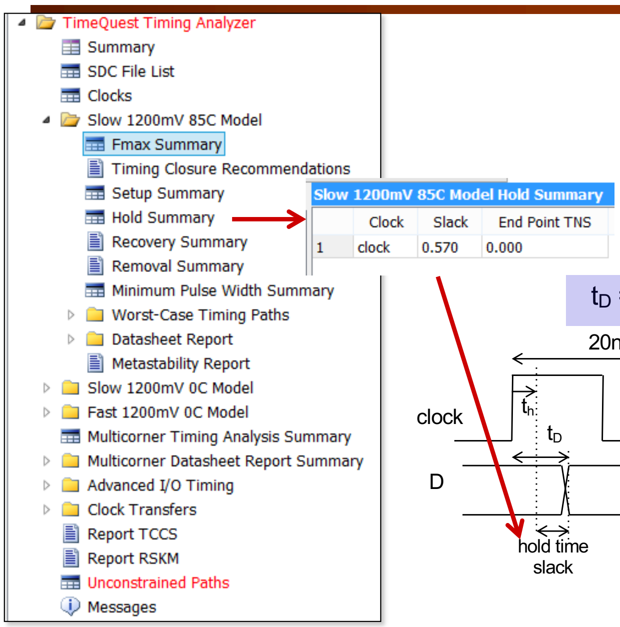
count + 1'b1 evaluated immediately after +ve edge of clock

PYKC 5 Dec 2023 EE2 Circuits & Systems Lecture 12 Slide 14

For this circuit, it also reports the timing slack. We are running the clock at 20ns period or 50MHz. Then the setup time slack is 17.721ns. That is D settles to its final value 17.721ns earlier than it is required.

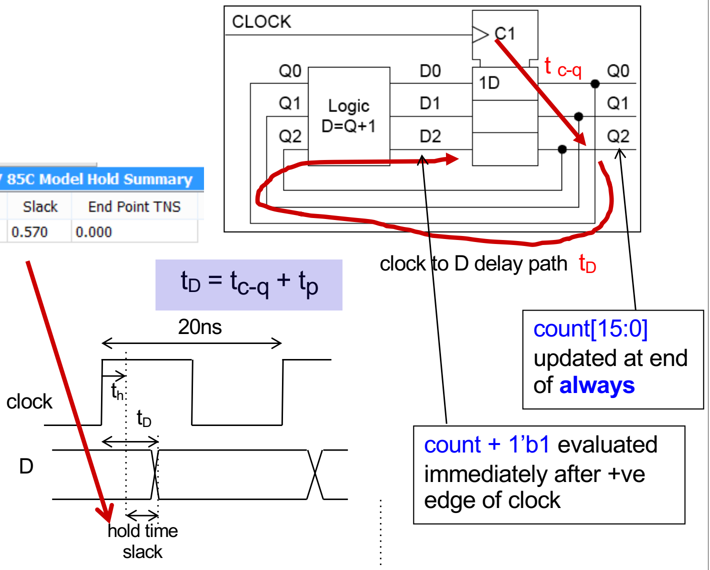
Slack time is a measure of the margin you have before the circuit stops working reliability. (Values will be different for MAX10 FPGA this year.)

TimeQuest Report (2) – Hold Summary



Slow 1200mV 85C Model Hold Summary

	Clock	Slack	End Point TNS
1	clock	0.570	0.000



$t_D = t_{c\text{-}q} + t_p$

count[15:0] updated at end of **always**

count + 1'b1 evaluated immediately after +ve edge of clock

PYKC 5 Dec 2023 EE2 Circuits & Systems Lecture 12 Slide 15

Hold time slack is reported here to be 0.57ns.